Sub A V

1. A method by which more than one client program connected to a network stores the same data item on a storage device of a data repository connected to the network, the method comprising:

encrypting the data item using a key derived from the content of the data item;

determining a digital fingerprint of the data item; and

storing the data item on the storage device at a location or locations associated with the digital fingerprint.

2. The method of claim 1 further comprising testing for whether a data item is already stored in the repository by comparing a digital fingerprint of the data item to digital fingerprints of data items already in storage in the repository.

3. The method of claim 2 wherein the same digital fingerprint is used for storing the data item on the storage device and for testing whether a data item is already stored in the repository.

4. The method of claim 1 wherein the encrypting of the data item is performed by the client prior to transmitting the data item to the storage device.

5. The method of claim 4 further comprising encrypting the key and storing the encrypted key on the storage device or on another storage device connected to the network.

6. The method of claim 5 wherein a client or user specific key is used to encrypt the key derived from the content of the data item.

7. The method of claim 1 wherein the key derived from the content of the data item is the same for all instances of the data item stored in the repository.

8. The method of claim 1 wherein users of the method are grouped into families, and the key derived from the content of the data item is the same for all instances of the data item

stored in the repository by users in the same family, but may be different for users in different families.

9. The method of claim 2 wherein one or more additional copies or other forms of redundant information about the data items is stored on the storage device or on other storage devices connected to the network for data integrity, availability, or accessibility purposes and not to provide separate storage of the data item for different client programs.

10. The method of claim 1 further comprising associating the data item with each of a plurality of access-authorization credentials, each of which is uniquely associated with a particular user or client program.

11. The method of claim 2 further comprising associating the data item with each of a plurality of access-authorization credentials, each of which is uniquely associated with a particular user or client program.

12. The method of claim 10 wherein the associating of the data item with each of a plurality of access-authorization credentials comprises storing a plurality of named objects, each named object comprising information representative of the data item paired with information representative of one of the access-authorization credentials.

13. The method of claim 12 wherein the information representative of the data item is a digital fingerprint.

14. The method of claim 12 wherein the information representative of the access-authorization credential is a cryptographic hash of all or part of the access-authorization credential.

15. The method of claim 14 wherein the cryptographic hash is an access identifier that uniquely identifies the data item for a particular user or client program.

16. The method of claim 12 wherein the named object is a data structure created by the client program.

17. The method of claim 12 wherein the named object is a data structure created by a server program acting on behalf of the repository.

18. The method of claim 12 further comprising a client replacing an existing version of a named object with a new version of that named object, by replacing the existing association with a data item stored on the storage device with a new association.

19. The method of claim 12 further comprising a client retrieving a data item by accessing a named object using an access-authorization credential to select the named object, and using the contents of the named object to determine the location of the data item on the storage device.

20. The method of claim 12 wherein the named objects further comprise version information associating different data items with different versions of the named object.

21. The method of claim 20 wherein a backup of data items stored on the storage device is accomplished by preserving copies of the current versions of named objects in existence at the time of the backup.

22. The method of claim 1 wherein records are kept of the association between data items and names in order to define named objects, and wherein data items recorded as being associated with named objects are not deleted from the repository, and wherein named objects are backed up by preserving copies of the named object records in existence at the time of the backup.

23. The method of claim 21 or 22 wherein a plurality of backups are made at spaced time intervals.

24. The method of claim 21 or 22 wherein the backup is accomplished by declaring that after a prescribed moment in time a new version of each named object will be created the first time that a new data item is associated with it.

25. The method of claim 24 wherein the prescribed moment in time is determined separately for each named object.

26. The method of claim 22 wherein named objects are preserved by creating a new version of each named object each time that a new data item is associated with it.

27. The method of claim 26, wherein versions of named objects that are deemed unnecessary are deleted.

28. The method of claim 27, wherein the determination of which versions of a named object to delete is based in whole or in part on the times at which the versions were created, and the intervals between these times.

29. The method of claim 20 further comprising preparing a digital time stamp of a plurality of named objects to allow a property of these named objects to be proven at a later date.

30. The method of claim 29 wherein a random or other difficult to guess element is incorporated into the time stamp hash for each named object, to prevent the property from being proven if this element is deleted.

31. The method of claim 12 further comprising determining that a data item stored on the storage device is not referenced by any named object, and reusing the storage space used to store the unreferenced data item.

32. The method of claim 12 further comprising altering one or more properties or parameters associated with an access-authorization credential to change the access rights of a client or user to the data item referenced by that credential.

33. The method of claim 2 further comprising a challenge step to ascertain that the client has the full data item.

34. The method of claim 33 wherein the challenge step comprises requiring that the client attempting to store a data item provide correct answers to inquiries as to the content of portions of the data item, or inquiries that require knowledge of this content.

35. The method of claim 34 wherein the data item content on which the challenge is based is selected with a degree of randomness.

36. The method of claim 2 wherein depositors use the client to store data items in the repository, and at least some depositors are required to provide identification.

37. The method of claim 36 wherein rules for when a depositor must provide identification are selected in order to discourage unlawful distribution of access to the data item.

38. The method of claim 37 wherein there is a greater degree of user identification or a higher likelihood that user identification will be required when the data item being stored by the depositor has been indicated to be shareable with other users.

39. The method of claim 37 wherein for a class of data items the items may only be shared if the depositor has provided adequate identification.

40. The method of claim 38 or 39 wherein identity information about the depositor is made available to anyone able to access the data item, to discourage unlawful sharing.

41. The method of claim 40 wherein the identity information is stored in an encrypted form that the depositor and users subsequently accessing the shared data item can both read.

42. The method of claim 41 wherein the repository is not able to decrypt the identity information about the depositor.

43. The method of claim 37 wherein the identity of some users has not been well verified, but restrictions are placed on sharing of data items deposited by such poorly verified users.

44. The method of claim 43 further comprising limiting access to data items deposited by a poorly verified user.

45. The method of claim 44 wherein the limited access is provided by limiting the aggregate bandwidth provided for such accesses.

46. The method of claim 44 wherein the limited access is provided by limiting the number of simultaneous accesses to the data items.

47. The method of claim 2 wherein the client has a directory structure for the data items, the data items are stored in the repository, and the directory structure is not evident to the repository maintainers.

48. The method of claim 2 wherein the client program using the repository is a mirroring program which determines which data items to deposit in the repository, and wherein that determination is based at least in part on the result of a comparison of digital fingerprints establishing that certain data items are not in the repository.

49. The method of claim 48 wherein mirroring software is downloaded to the client using a bootstrap process, wherein a small bootstrap program is downloaded and executed,

and the bootstrap program manages download and installation of the remainder of the mirroring software.

50. The method of claim 48 wherein the default for deciding what data items to mirror is to mirror all or substantially all data items.

51. The method of claim 48 wherein the mirroring comprises making a determination of which data items need to be transmitted to the repository, and wherein that determination is based primarily on a comparison of digital fingerprints for data items at the client and data items in the repository.

52. The method of claim 10 wherein the access-authorization credential is determined in part by computing a hash involving elements of the pathname for a file on the client computer.

53. The method of claim 52 wherein the path name hash is made unique to a client by introducing a reproducible but randomly chosen element into it.

54. The method of claim 12 wherein a data item is represented as a composite of data-items, and the component data-items are separately deposited in the repository.

55. The method of claim 54 wherein lists of fingerprints for data-items making up a composite data-item are deposited as an index data item, which can be given an object-name and used for obtaining access to any of the component data-items.

56. The method of claim 55 wherein a proof-of-deposit is returned for each component deposit, and some or all of the proofs are presented when the index data item is given an object-name.

57. The method of claim 56 wherein, when transmitting a composite data-item, the client uses fingerprints to avoid retransmitting components following loss of communication.

58. The method of claim 57 wherein the index data-item is encrypted with a key that is only made available to the repository at the moment of access.

59. The method of claim 55 wherein an email message is broken up into component items in such a manner that the individual attachments are separate component data-items.

60. The method of claim 15 wherein the physical location at which information about named-objects is stored is based on access identifiers, to introduce reproducible pseudorandomness into the physical locations of the named-object data.

61. The method of claim 1 wherein the fingerprints are determined from the data items, and this process produces randomly distributed numbers which can be used to introduce reproducible pseudorandomness into the physical locations of the data items.

62. The method of claim 2 wherein an access identifier is formed to provide proof of ownership of the data item stored in the repository, the access identifier is formed by producing a one-way hash including item-identifying information chosen by the client program to identify the data item, and the one-way hash cannot be reversed to permit the repository to discover the identity of the client program or user.

63. The method of claim 62 wherein the item-identifying information is associated with the data item on the client.

64. The method of claim 63 wherein the item-identifying information is derived at least in part from the path name of the data item on the client.

65. The method of claim 62 wherein user-identifying information is provided to the repository as part of the access-authorization credential.

66. The method of claim 65 wherein at least some access-authorization credentials can be transferred between users without the use of the repository.

67. The method of claim 65 wherein at least one class of users is not permitted to transfer access using access-authorization credentials.

68. A method by which more than one client program connected to a network stores the same data item on a storage device of a data repository connected to the network, the method comprising:

determining a digital fingerprint of the data item;

testing for whether the data item is already stored in the repository by comparing the digital fingerprint of the data item to the digital fingerprints of data items already in storage in the repository; and

challenging a client that is attempting to deposit a data item already stored in the repository, to ascertain that the client has the full data item.

69. The method of claim 68 wherein the repository gives the client a deposit receipt which allows the user to prove that the deposit occurred.

70. The method of claim 68 wherein the challenging comprises requiring that the client provide correct answers to inquiries as to the content of portions of the data item, or inquiries that require knowledge of this content.

71. The method of claim 70 wherein the data item content on which the challenge is based is not easily predicted by the user or client program.

72. The method of claim 70 wherein the data item content on which the challenge is based can be determined by the client program without the aid of the repository.

73. The method of claim 68 wherein future access to the data item deposited is provided by creating an access-authorization credential which can be presented at a later time to prove that the challenge has been met for that data item.

74. The method of claim 73 wherein each access authorization credential is uniquely associated with a access owner.

75. The method of claim 73 wherein each access authorization credential includes information sufficient to identify the access owner.

76. The method of claim 73 wherein the access authorization credential includes a fingerprint.

77. The method of claim 73 wherein the access authorization credential is associated with a fingerprint in the repository.

78. The method of claim 76 or 77 wherein the fingerprint is different from the fingerprint used for testing whether the data item is already stored in the repository.

79. The method of claim 73 wherein the access authorization credential is associated directly with the data-item or with a record in the repository that is associated with the data-item.

80. The method of claim 79 wherein the record in the repository with which the access authorization credential is associated is an access identifier that is associated with the credential by computation of a one way hash function.

81. The method of claim 80 wherein the access identifier is stored in the repository and is compared with a later hash of an access authorization credential to verify access permission to a named object.

82. The method of claim 73 wherein the access authorization credential may include information sufficient to respond to a challenge.

83. The method of claim 73 wherein the access authorization credential includes data proof information created during a challenge process that is sufficient to prove to the repository that the challenge was passed.

84. The method of claim 83 wherein the data proof information comprises the actual challenge response, so that it can be directly verified against the data-item.

85. The method of claim 73 wherein at least some access-authorization credentials can be transferred between users without the aid of the repository.

86. The method of claim 85 wherein the usage of some access authorization credential is restricted for at least one class of access owners.

87. The method of claim 86 wherein the access authorization credential is only usable by the access owner.

88. The method of claim 86 wherein the aggregate bandwidth available to all users of the access authorization credential is limited.

89. The method of claim 68 wherein at the time of deposit at least some data items are associated with a minimum expiration time.

90. The method of claim 89 wherein at least some data items that expire are removed and their storage space reused.

91. The method of claim 90 wherein the repository keeps track of which access owners have deposited a given data item.

92. The method of claim 91 wherein upon an access owner informing the repository that a data item is no longer needed, the data item is deleted or the expiration of the data item is accelerated.

93. The method of claim 92 wherein the repository truncates the list of depositors associated with a data-item, and never accelerates the expiration of this data item.

94. The method of claim 68 further comprising encrypting the data item using a key derived from the content of the data item.

95. The method of claim 94 wherein the encrypting of the data item is performed by the client prior to transmitting the data item to the storage device.

96. The method of claim 94 further comprising encrypting the key and storing the encrypted key on the storage device or on another storage device connected to the network.

97. The method of claim 96 wherein a client or user specific key is used to encrypt the key derived from the content of the data item.

98. A method by which more than one client program connected to a network stores the same data item on a storage device of a data repository connected to the network, the method comprising:

determining a digital fingerprint of the data item;

storing the data item on the storage device at a location or locations associated with the digital fingerprint;

associating the data item with each of a plurality of access-authorization credentials, each of which is uniquely associated with an access owner; and

preparing a digital time stamp of a plurality of records associating data-items and credentials, to allow a property of these records to be proven at a later date.

99. The method of claim 98 wherein preparing the digital time stamp comprises forming a time stamp hash, and wherein a difficult to guess or random element is incorporated into the time stamp hash, to prevent the property from being proven if this element is deleted.

100. The method of claim 98 wherein all data items in the repository are time stamped if they remain in the depository for a sufficiently long time period.

101. A method for quantifying the degree of uniqueness of an indicated data item in a repository of data items stored on a storage device at locations associated with their digital fingerprints, the method comprising:

creating access-authorization credentials which permit users or clients to access data-items that they have deposited; and

determining (or approximating) the number of users with access authorization credentials for the indicated data item.

102. The method of claim 101 wherein the data item is a portion of the body of an e-mail message, and the method is used to determine the relative uniqueness of the portion of the e-mail message in a large population of e-mail messages to determine the likelihood that the e-mail is spam.

103. The method of claim 101 wherein a decision as to whether a data item is a virus is made by comparing the relative uniqueness of both the data item and other data items associated with the same application.

104. The method of claim 101 further comprising collecting and providing usage statistics based on the degree of uniqueness of data items in the repository.

105. The method of claim 104 wherein the usage statistics are configured to provide marketing penetration information on the data item.

106. A method by which more than one client connected to a network stores the same data item on a storage device of a data repository connected to the network, the method comprising:

determining a digital fingerprint of the data item;

testing for whether a data item is already stored in the repository by comparing the digital fingerprint of the data item to the digital fingerprints of data items already in storage in the repository; and

associating with a data item an informational tag which may be read by at least some client programs.

107. The method of claim 106 wherein the informational tag indicates at least one of the following: whether the data item contains spam, whether the data item contains or is a virus, whether the data item is copyrighted, by whom the data item is copyrighted, what royalty payment is due for the copyright.

108. The method of claim 107 further comprising the process of collecting royalties or other payments for use of a copyright on a data item based on the indication of whether a data item is copyrighted.

109. The method of claim 108 wherein the process enables voluntary payment of such royalties or payments.

110. The method of claim 106 further comprising encrypting the data item using a key derived from the content of the data item.

111. The method of claim 110 wherein at least some of the tags are encrypted using the same key as for each data item, so that users with the data item can read the informational contents of the tag.

112. A method by which more than one client connected to a network may store the same data item on a storage device of a data repository connected to the network, and

wherein there is a public data repository and a private data repository, the method comprising:

determining a digital fingerprint of the data item;

testing for whether a data item is already stored in the public repository by comparing the digital fingerprint of the data item to the digital fingerprints of data items already in storage in the public repository; and

if the data item is present in the public repository, creating an access authorization credential for the public repository associating the client with the data item and relying on storage of the data item in the public repository; and if the data item is not present in the public repository, creating an access authorization credential for the private repository and relying on storage of the data item in the private repository.

113. The method of claim 112 wherein the client creates an access authorization credential for the data item exclusively either in the public or the private repository.

114. The method of claim 2 wherein the data items are widely circulated non-electronic media such as books or music, and the method further comprises converting the widely circulated non-electronic media to a standardized electronic version;

storing the standardized electronic version as a data item in the repository;

promoting the availability of the standardized electronic version to users with the right to have access, whereby the likelihood of the data repository storing multiple, slightly-different electronic versions of the non-electronic media is reduced.

115. A method by which a client connected to a network over a lower speed connection may provide higher speed access to a data item for application processing than is possible over the relatively low speed connection to the network, the method comprising:

determining a digital fingerprint of the data item;

testing for whether the data item is already stored in a repository by comparing the digital fingerprint of the data item to digital fingerprints of data items already in the repository;

only if the data item is not already in the repository, transferring the data item over the lower speed connection from the client to the repository, the repository being connected to the network over a higher speed connection than the client;

making a higher speed connection between an application server and the data repository;

executing an application on the application server to process the data item stored on the data repository;

returning at least some of the processed data to the client across the lower speed connection.

116. The method of claim 115 wherein one or both of the data transfers to and from the client are conducted in the background while other applications are running on the client.

117. A method by which multiple clients browse content on a network such as the Internet, the method comprising:

each of the multiple clients accessing content on the network via one or more proxy servers;

determining the digital fingerprint of an item of content passing through the proxy server;

storing the item of content in a content repository connected to the proxy server at a location associated with the digital fingerprint;

testing for whether a content data item is already stored in the repository by comparing the digital fingerprint of the content data item to the digital fingerprints of content data items already in storage in the repository;

associating a content data item already stored in the repository with an access authorization credential uniquely associated with an access owner.

118. The method of claim 117 wherein the data repository saves substantially all content browsed by the clients, thereby preserving the content after it has been altered or removed from the network.

119. The method of claim 118 further comprising granting search engines access to the stored content data items or to information about the number of times that data items have been accessed or how recently the data items have been accessed

120. A method by which a plurality of clients connected to a network store the same broadcast data on a storage device of a data repository connected to the network, wherein the broadcast data comprises a sequence of frames or other fragments, the method comprising:

determining a digital fingerprint of each fragment;

testing for whether the fragment is already stored in the repository by comparing a digital fingerprint of the fragment to digital fingerprints of fragments and other data items already in storage in the repository;

having only the client or clients that determine that a fragment is not stored in the repository transmit the fragment to the repository;

whereby because all but one or a small number of clients will not have to transmit the fragment to effect storage of the fragment in the repository, most of the clients are able to store the broadcast data in the repository without actually transmitting a significant fraction of the data to the repository.

121. The method of claim 120 wherein the broadcast data is video and the fragments are frames of video.

122. A method of encrypting a bit-string using cellular automata, comprising

dividing the bit-string into segments in which at least some bits in each segment are considered to be homologous;

transforming disjoint groups of homologous bits by applying a state-permutation operation separately to each group; and

changing which bits are considered to be homologous and repeating the process.

123. The method of claim 122 wherein the arrangement of bits into segments can be expressed as having a spatial interpretation, and the spatial origin of each segment is shifted

in a manner determined by an encryption key, with bits in different segments that have the same spatial coordinates considered to be homologous.

124. The method of claim 123 wherein an encryption key is used to determine what state-permutation operation is applied to each group of homologous bits in each step.

125. The method of claim 48 wherein the aforesaid steps of the method provide a mirroring capability for a personal computer, and mirroring software with instructions for carrying out the aforesaid steps is preconfigured on the personal computer upon purchase.

126. The method of claim 48 wherein the aforesaid steps of the method provide a mirroring capability for a personal computer, and mirroring software for carrying out the method is initially configured to mirror essentially all data on the user's computer.

127. The method of claim 48 wherein the aforesaid steps of the method provide a mirroring capability for a wireless network device.

128. A method for selling a backup service for backing up or mirroring data on a client computer, the method comprising:

accepting an unlimited amount of backup or mirroring data from a plurality of client computers, and storing the data in one or more repositories to which the client computers are connected via a network, for free or at a charge substantially less than sufficient to cover the cost of operating the backup service;

charging a substantial fee, greater than the fee charged for accepting the data, for recovery of the data from the repositories.

129. The method of claim 128 wherein the fee charged for recovery is greater when the recovered data is provided quickly, either by express delivery of media containing the data or by delivery over a high-speed data connection.

130. The method of claim 128 wherein recovery of data over a slow-speed data connection is provided at no fee or at a charge substantially less than sufficient to cover the cost of operating the backup service.

131. The method of claim 128, 129, or 130 wherein data coalescence using digital fingerprints is used to reduce the amount of data transmitted and stored during backup or mirroring.

132. The method of claim 128 wherein a charge is made to third parties for high-speed network access to the client data resident on the repositories.

133. The method of claim 68 wherein records are kept of the association between data items and names in order to define named objects, and wherein data items recorded as being associated with named objects are not deleted from the repository, and wherein named objects are backed up by preserving copies of the named object records in existence at the time of the backup.

134. The method of claim 68 wherein a backup of data items stored on the storage device is accomplished by preserving copies of the current versions of named objects in existence at the time of the backup.

135. The method of claim 133 or 134 wherein a plurality of backups are made at spaced time intervals.

136. The method of claim 133 or 134 wherein the backup is accomplished by declaring that after a prescribed moment in time a new version of each named object will be created the first time that a new data item is associated with it.

137. The method of claim 136 wherein the prescribed moment in time is determined separately for each named object.

138. The method of claim 133 wherein named objects are preserved by creating a new version of each named object each time that a new data item is associated with it.

139. The method of claim 138 wherein versions of named objects that are deemed unnecessary are deleted.

140. The method of claim 139 wherein the determination of which versions of a named object to delete is based in whole or in part on the times at which the versions were created, and the intervals between these times.

141. The method of claim 68 wherein depositors use the client to store data items in the repository, and at least some depositors are required to provide identification.

142. The method of claim 141 wherein rules for when a depositor must provide identification are selected in order to discourage unlawful distribution of access to the data item.

143. The method of claim 142 wherein there is a greater degree of user identification or a higher likelihood that user identification will be required when the data item being stored by the depositor has been indicated to be shareable with other users.

144. The method of claim 142 wherein for a class of data items the items may only be shared if the depositor has provided adequate identification.

145. The method of claim 143 or 144 wherein identity information about the depositor is made available to anyone able to access the data item, to discourage unlawful sharing.

146. The method of claim 145 wherein the identity information is stored in an encrypted form that the depositor and users subsequently accessing the shared data item can both read.

147. The method of claim 146 wherein the repository is not able to decrypt the identity information about the depositor.

148. The method of claim 143 wherein the identity of some users has not been well verified, but restrictions are placed on sharing of data items deposited by such poorly verified users.

149. The method of claim 148 further comprising limiting access to data items deposited by a poorly verified user.

150. The method of claim 149 wherein the limited access is provided by limiting the aggregate bandwidth provided for such accesses.

151. The method of claim 149 wherein the limited access is provided by limiting the number of simultaneous accesses to the data items.

152. The method of claim 73 wherein the access-authorization credential is determined in part by computing a hash involving elements of the pathname for a file on the client computer.

153. The method of claim 152 wherein the path name hash is made unique to a client by introducing a reproducible but randomly chosen element into it.